

5
200

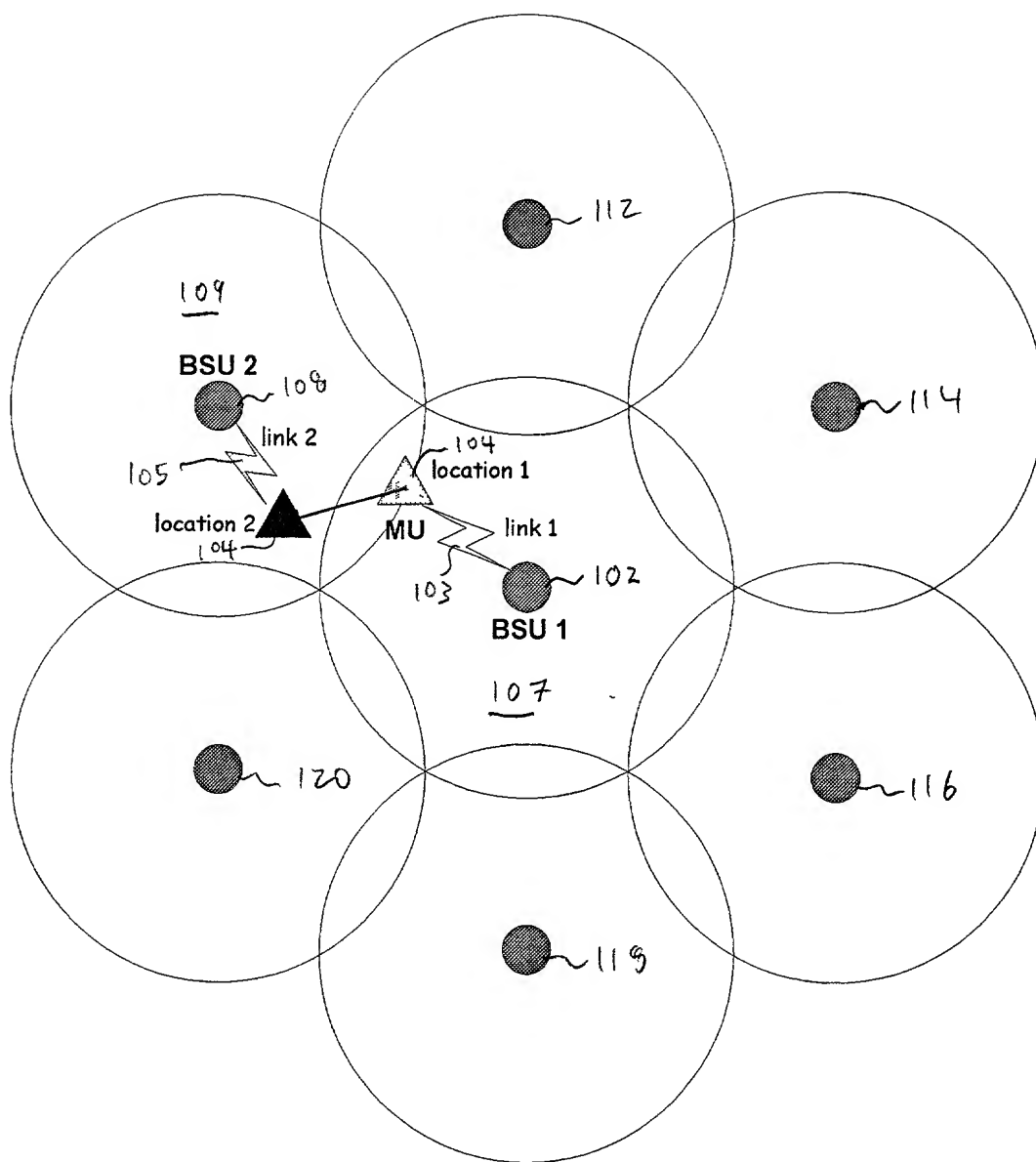


Fig 1

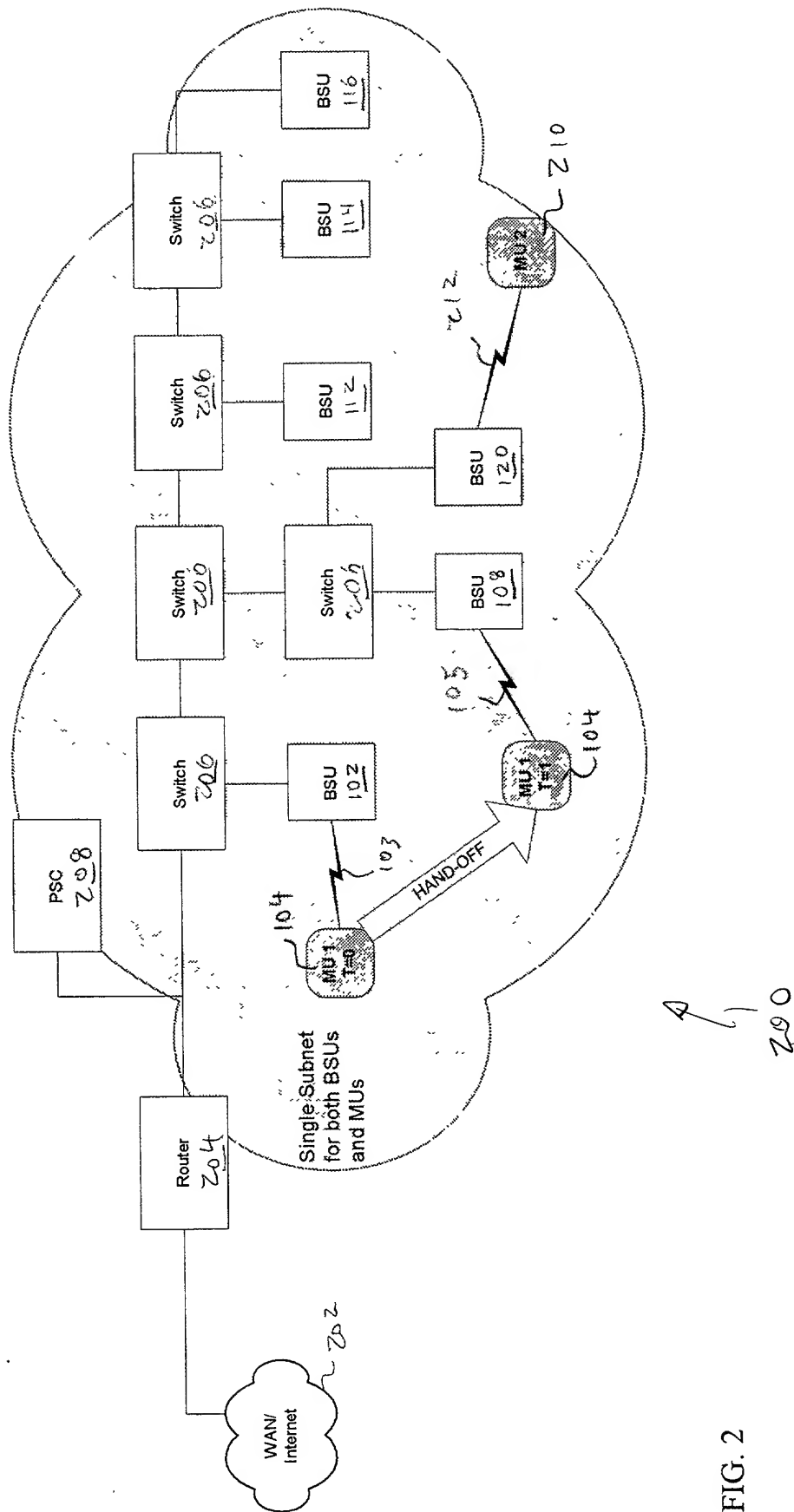
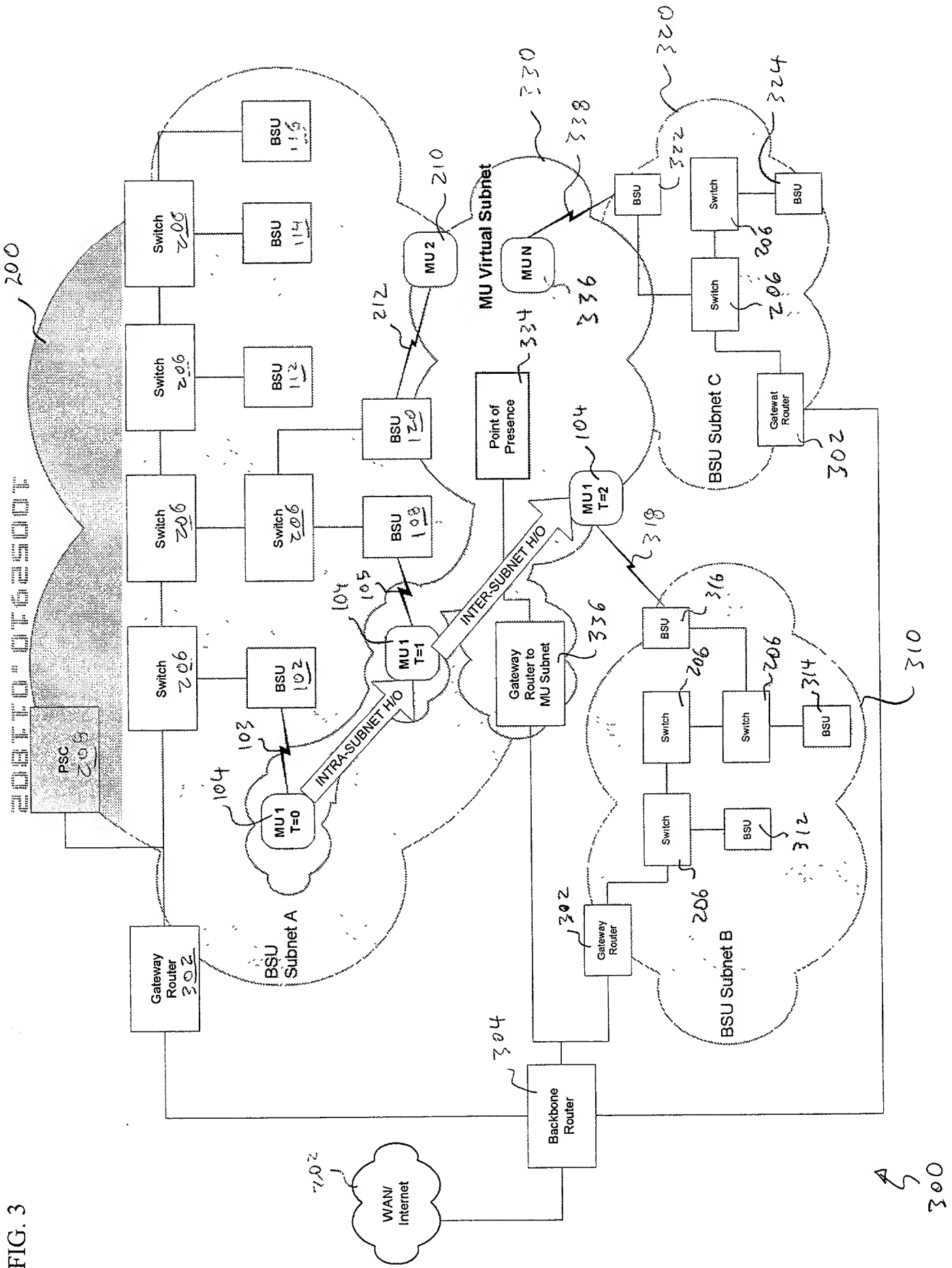


FIG. 2

FIG. 3



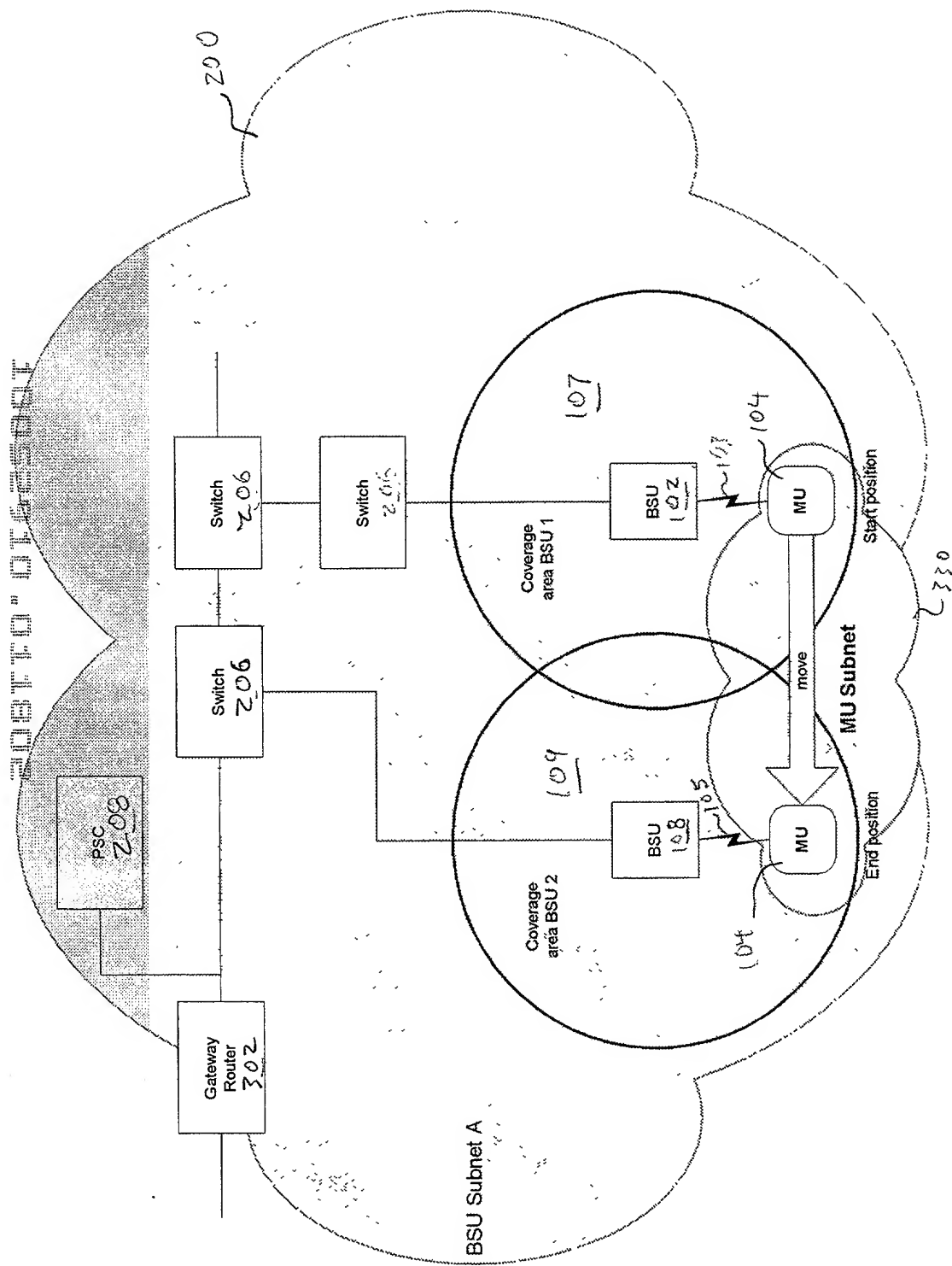


Figure 4 Intra-subnet Hand-off

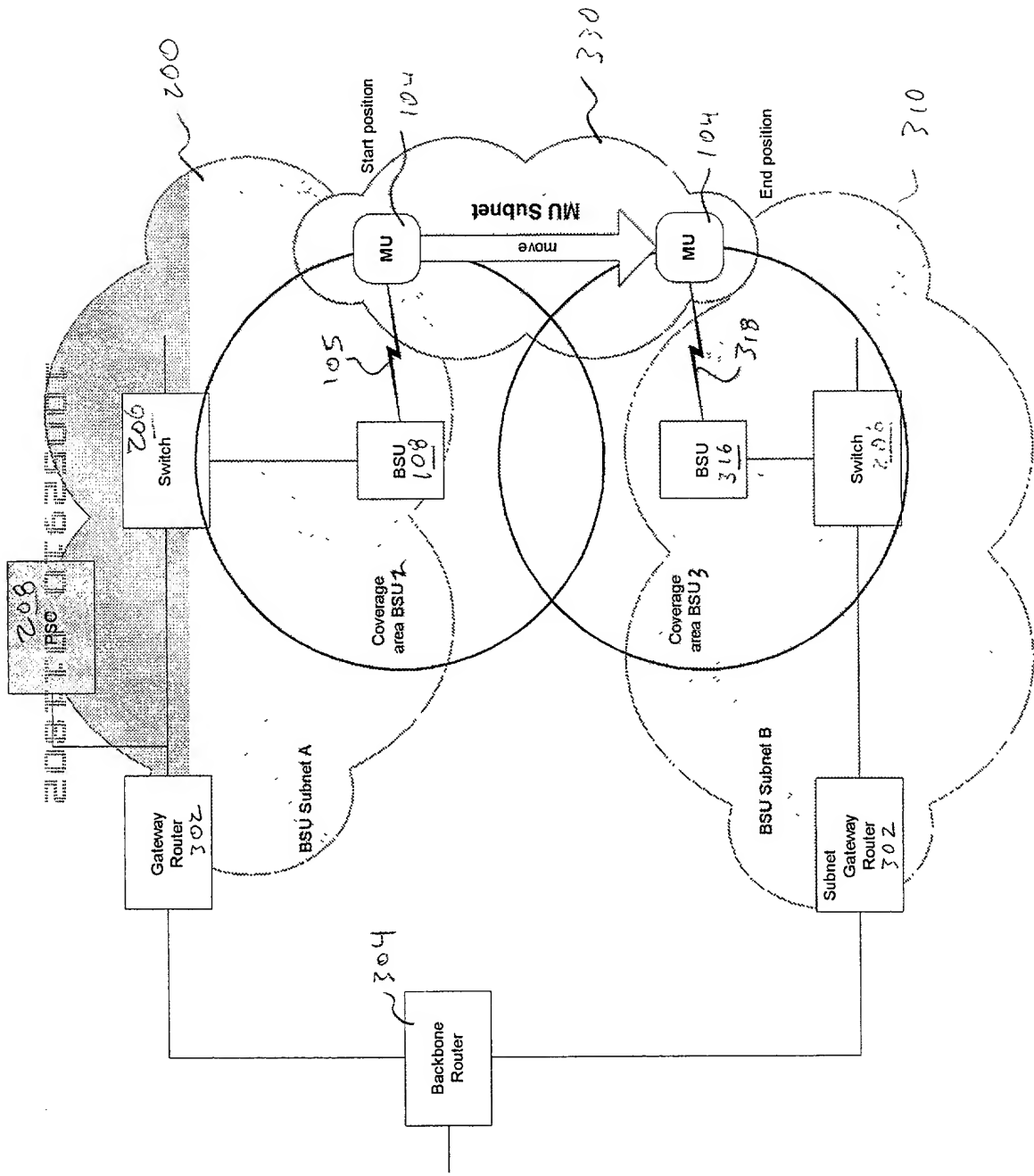


Figure 5 Inter-subnet Hand-off

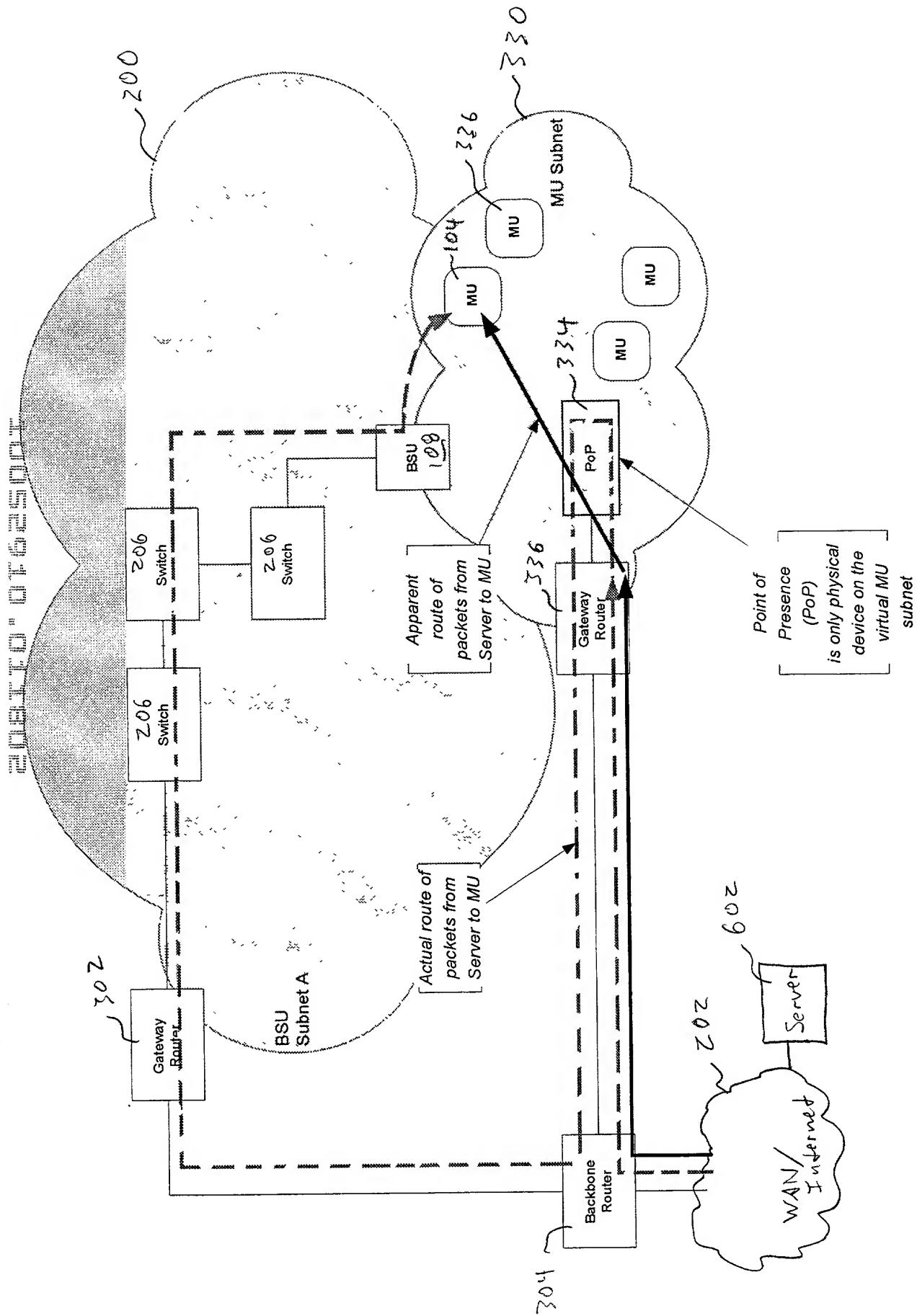
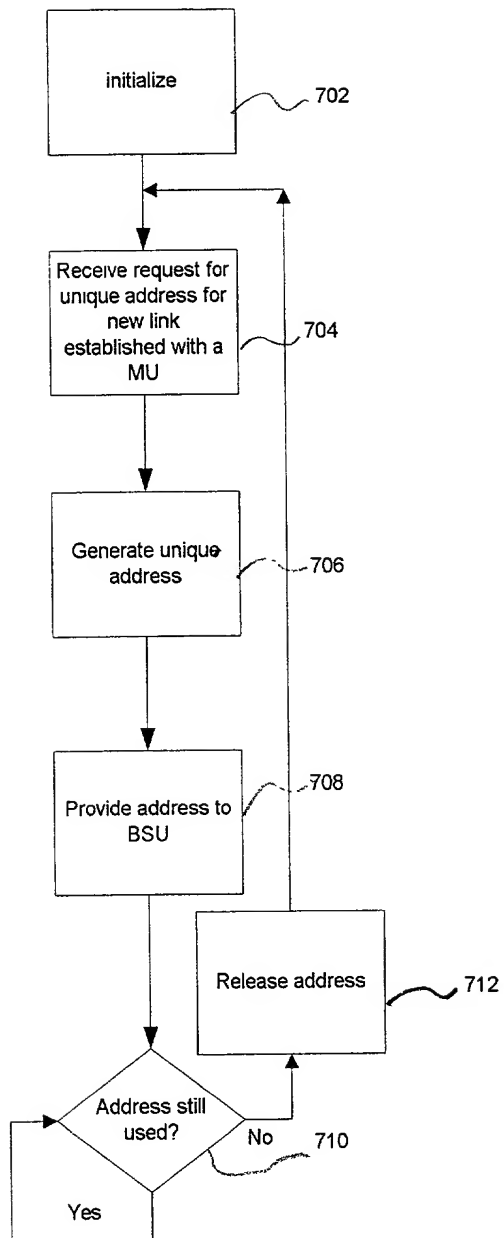


FIG. 6

2025-10-04 10:59:00



700

FIG. 7

900

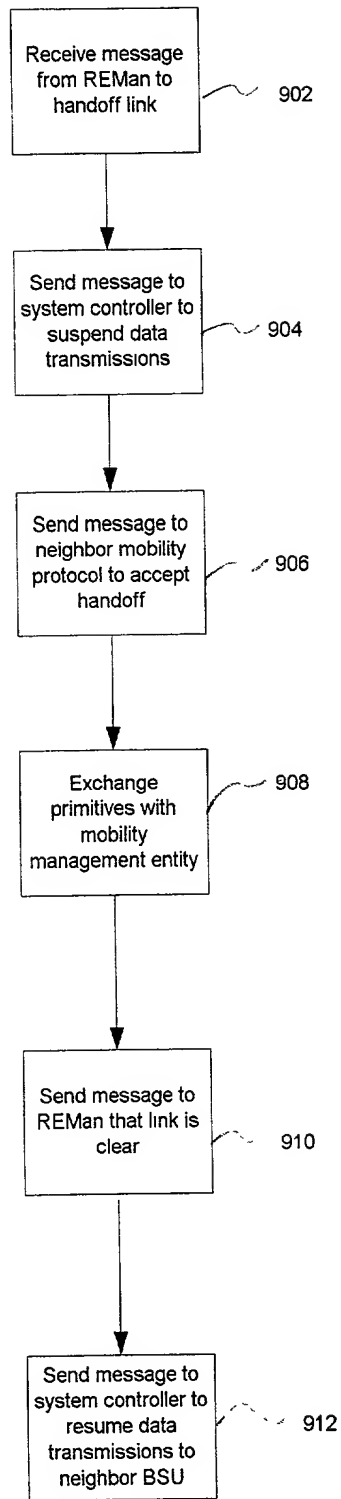


FIG. 9

Link Context Record

SET OF 0162500T

Virtual Bluetooth device address ("BD Addr")	<unique 48 bit address>	~1002
BSU system clock offset (CLK OFFSET)	<offset value>	~1004
Active Member Address for MU	<integer 0:7>	~1006
Encryption keys (optional)	<integer>	~1008
BSU's IP Address	<local IP Address>	~1010
Mode and timing parameters	<Mode: Sniff, Hold, Park>, <time>	~1012
Mobile Unit ID ("BD Addr")	<unique 48 bit address>	~1014
Channel ID	<16 bit value>	~1016
Link Initialization Time	<date, time>	~1018

Fig. 10

1000

1100

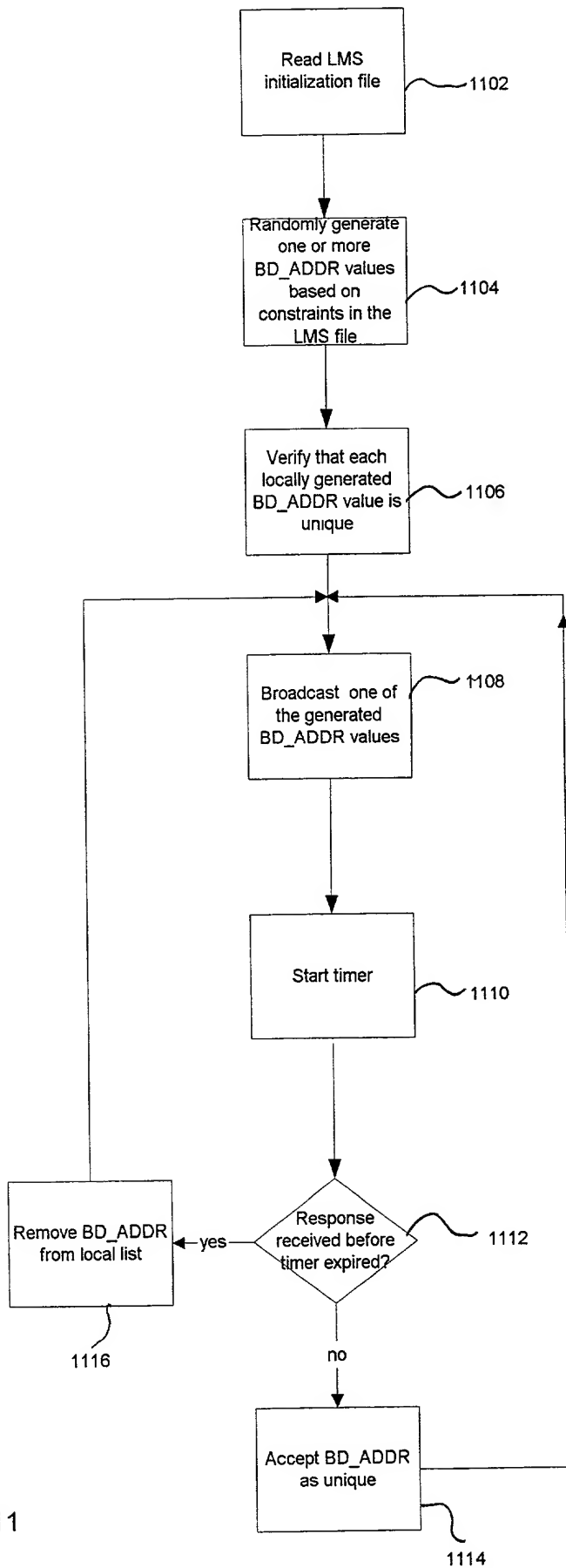


FIG. 11

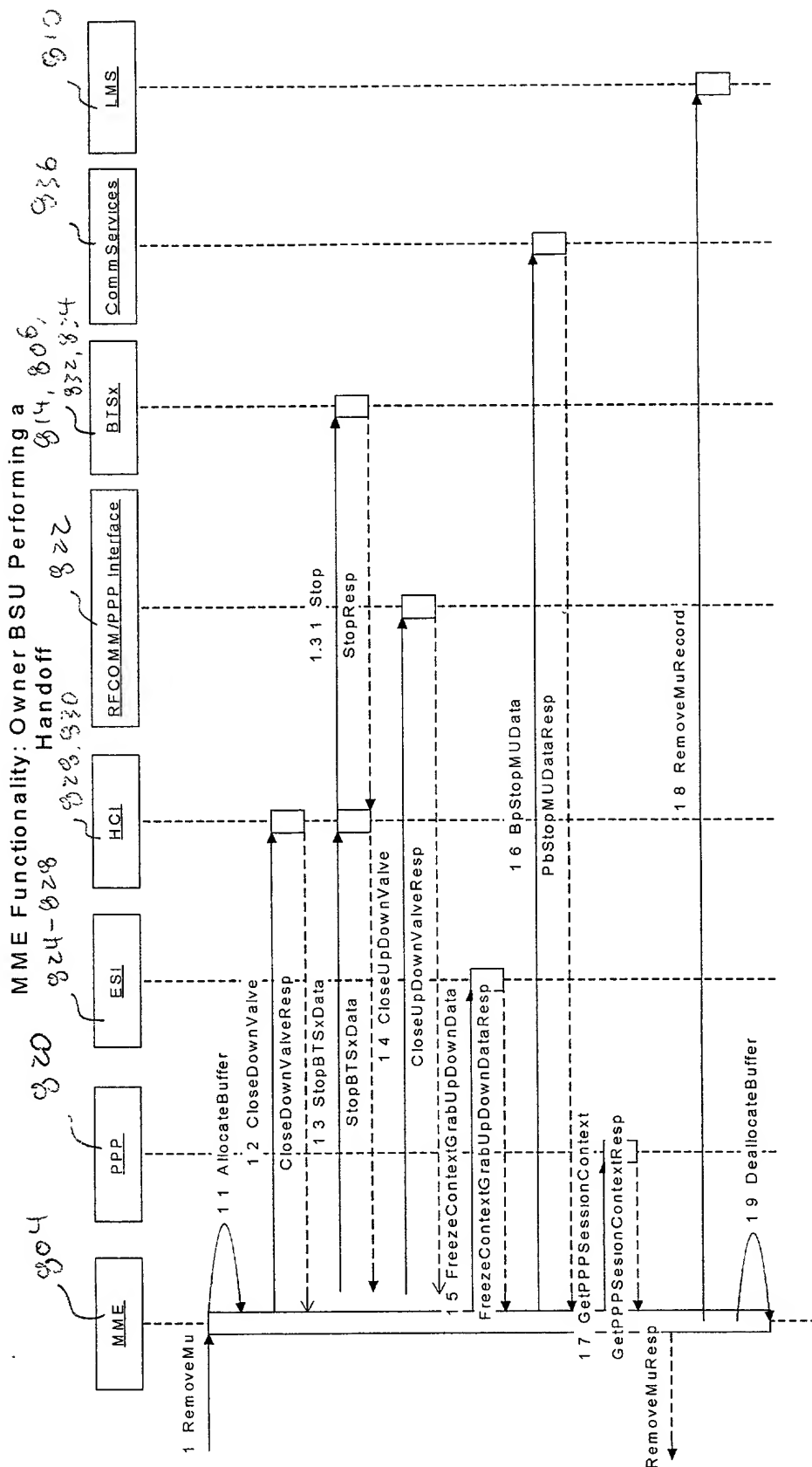


Figure 12: The MME on the owner BSU receives a RemoveMu call. That calls starts a chain of messages that will result in the buffered data flowing to the BTSx getting captured and sent to the target BSU. Along with that data will come the session context.

MME Functionality: Target BSU Receiving a Handoff

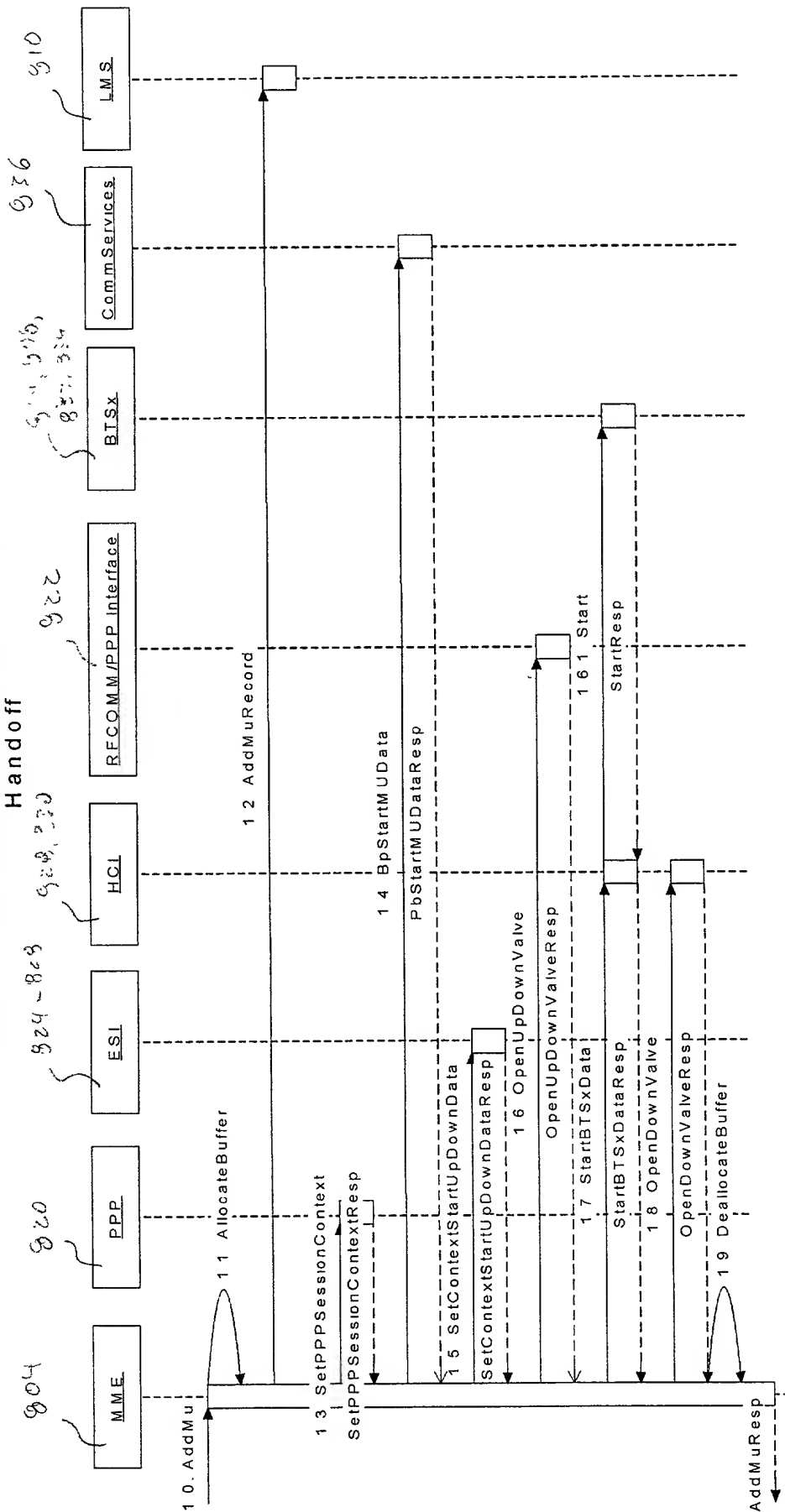
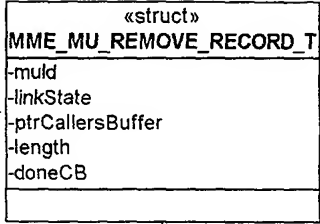
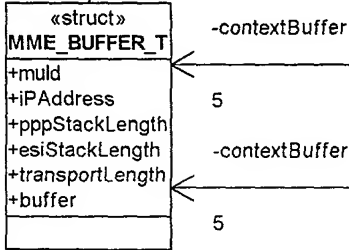
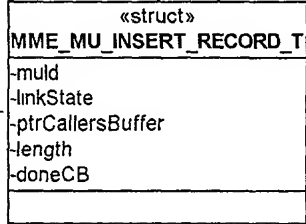


Figure 13: The MME gets its AddMu method called. This means that a MU session must be established. This includes any data that is being sent to the MU and the session context.

This structure contains the session context from the owner BSU to be transferred to the target BSU. This struct is defined in mme.h.



These are mme c housekeeping structures that form a table of MUs that are being handed-off. The MU id, the MU's handoff state with MME, the pointer to the callers buffer, the length of the callers buffer, and the callers callback are maintained in these structures.



F 14

[illegible]